IN THE CLAIMS:

1. (currently amended) A system including a multi-tier application architecture having a middletier, said system comprising:

a framework to mediate between an application and [[a]] the middletier, wherein the framework is configured to:

~~allows a~~ allow the middletier to execute an object fetched by [[an]] the application from a cache;

~~wherein,~~ when [[an]] the execution of the object ~~becomes stale~~ fails, ~~the framework~~ repeatedly ~~refreshes~~ refresh the object within a limited number of retries;

~~wherein,~~ when [[an]] the object refresh succeeds, ~~the framework returns~~ return the object to the cache and again ~~allows~~ allow the middletier to execute the object; and

~~wherein,~~ when [[an]] the object refresh does not succeed within [[a]] the limited number of retries, ~~the framework quits an~~ quit the application in fail-safe way.

2. (currently amended) The ~~multi-tier application architecture~~ system according to claim 1, wherein the framework is configured to allow a user [[can]] to specify the limited number of retries.

3. (currently amended) The ~~multi-tier application architecture~~ system according to claim 2, wherein [[a]] the framework is configured to allow the user [[can]] to specify a time interval between the retries.

4. (currently amended) The ~~multi-tier application architecture~~ system according to claim 1, wherein the framework has its operations visualized to a user.

5. (currently amended) The ~~multi-tier application architecture~~ system according to claim 1, further including[[:]] a watchdog configured to resume normal operations when the middletier crashes.

6. (currently amended) The ~~multi-tier application architecture~~ system according to claim 5, wherein the watchdog ~~recovers a~~ is configured to recover the middletier based on a result of periodical polling.

7. (currently amended) The ~~multi tier application architecture~~ system according to claim 5, wherein the watchdog ~~recovers a~~ is configured to recover the middletier based on notification from the framework.

8. (new) The system according to claim 1, wherein the framework comprises a logic controller, a detector, a refresher, and a quitter.

9. (new) A method of executing an application, said method comprising:

transmitting an object used by the application from a first tier to a second tier;

executing a logic program at the second tier, wherein the logic program corresponds to the transmitted object;

detecting an execution status of the logic program at the first tier, said detecting comprising:

detecting when the execution of the logic program fails such the object becomes stale;

repeatedly refreshing the object within a limited number of retries; and

if said refreshing succeeds, then returning the object to the first tier and transmitting a second object to the second tier from the first tier; and

if said refreshing does not succeed within the limited number of retries, then quitting the application in fail-safe way.

10. (new) A method in accordance with Claim 9 wherein transmitting an object used by the application further comprises transmitting the object from a cache within the first tier to the second tier.

11. (new) A method in accordance with Claim 10 wherein transmitting an object from a cache further comprises transmitting the object from the cache through a framework within the first tier to the second tier.

12. (new) A method in accordance with Claim 9 wherein detecting an execution status of the logic program at the first tier further comprises detecting an execution status of the logic program at a framework within the first tier.

13. (new) A method in accordance with Claim 9 further comprising, when the second tier crashes, resuming normal operation using a watchdog.

14. (new) A method in accordance with Claim 13 wherein resuming normal operation further comprises resuming normal operation based on periodical polling of the second tier.

15. (new) A method in accordance with Claim 13 wherein resuming normal operation further comprises recovering the second tier based on notification from a framework within the first tier.

16. (new) A computer program embodied on a computer readable medium, said computer program comprising a code segment that:

transmits an object used by an application from a first tier to a second tier;

executes a logic program at the second tier, wherein the logic program corresponds to the transmitted object;

detects an execution status of the logic program at the first tier, wherein the code segments are configured to detect by:

detecting when the object becomes stale;

repeatedly refreshing the object within a limited number of retries; and

if said refreshing succeeds, then returning the object to the first tier and transmitting a second object to the second tier from the first tier; and

if said refreshing does not succeed within the limited number of retries, then quitting the application in fail-safe way.

17. (new) A computer program in accordance with Claim 16 further comprising a code segment that prompts a user to specify the limited number of retries.

18. (new) A computer program in accordance with Claim 17 further comprising a code segment that prompts a user to specify a time interval between the each of the limited number of retries.

19. (new) A computer program in accordance with Claim 16 further comprising a code segment that, when the second tier crashes, resumes normal operation based on periodical polling of the second tier.

20. (new) A computer program in accordance with Claim 16 further comprising a code segment that, when the second tier crashes, recovers the second tier based on notification from a framework within the first tier.